АРИФМЕТИЧЕСКОЕ КОДИРОВАНИЕ ВЕРОЯТНОСТНЫХ ИСТОЧНИКОВ*

В. Н. Потапов

Институт математики им. С. Л. Соболева СО РАН
Новосибирск, пр. Академика Коптюга 4,
e-mail: vpotapov@math.nsc.ru

Введение

Арифметическое кодирование является одним из наиболее известных методов, применяемых для сжатия текстов. Математической моделью множества текстов является множество последовательностей, состоящих из букв некоторого алфавита, с заданной на этом множестве вероятностной мерой, т.е. с заданными вероятностями букв в различных контекстах. Кодированием называется инъективное преобразование текстов в двоичные последовательности. Основными критериями эффективности метода кодирования являются степень сжатия текста, объем памяти, который требуется для программной реализации метода, и трудоемкость кодирования и декодирования. Арифметическое кодирование является одним из методов, наилучшим образом сочетающих высокую степень сжатия текста с простотой операций кодирования и декодирования.

Идея арифметического кодирования, которая будет изложена ниже, была высказана П. Элайесом. В 1976 году Й. Риссанен предложил первый эффективный метод вычисления арифметического кода (см. [6, 7]). В дальнейшем возникло несколько методов, реализующих идею арифметического кодирования. Один из них, предложенный в работе [8], подробно описан в настоящей статье. Изложение подобного алгоритма на русском языке, а также оценки его трудоемкости можно найти в

*Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (код проекта 99-01-00531) и ФЦП "Интеграция" (объединенный проект АО 110)

статье Б. Я. Рябко и А. Н. Фионова [5].

1. Основные определения

Пусть $A = \{a_1, ..., a_k\}$ — конечный алфавит. Бесконечные в обе стороны последовательности букв алфавита A образуют пространство элементарных событий A^∞. Говорят, что задан некоторый *источник* сообщений S в алфавите A, если определены вероятности всех событий вида $\{x \in A^\infty : x_{i_1} = a_{j_1}, \ldots, x_{i_n} = a_{j_n}\}$, где i_t — целые числа и $a_{j_t} \in A$. Другими словами, источник сообщений определен, если известна вероятность того, что сообщение $x \in A^\infty$ содержит в i_1-й позиции букву a_{j_1}, в i_2-й позиции — a_{j_2} и т. д. Источник называется *стационарным*, если вероятности событий не зависят от сдвига последовательности, т. е.

$$Pr\{x_{i_1} = a_{j_1}, \ldots, x_{i_n} = a_{j_n}\} = Pr\{x_{t+i_1} = a_{j_1}, \ldots, x_{t+i_n} = a_{j_n}\}.$$

для всех целых t. Для стационарных источников можно определить вероятность слова $a_{j_1}, \ldots, a_{j_n}$ равенством $p(a_{j_1}, \ldots, a_{j_n}) = Pr\{x_{t+1} = a_{j_1}, \ldots, x_{t+n} = a_{j_n}\}$, где t — произвольное целое число. Стационарный источник называется *источником без памяти* или *источником Бернулли*, если вероятность произвольной буквы $a \in A$ не зависит от соседних букв, т. е. $p(a_{j_1}, \ldots, a_{j_n}) = p(a_{j_1}) \ldots p(a_{j_n})$.

Энтропией стационарного источника S называется величина

$$H(S) = -\lim_{n\to\infty} \frac{1}{n} \sum_{x \in A^n} p(x) \log p(x) \tag{1}$$

(здесь и далее $\log = \log_2$). Известно (см., например, [1, 2, 3, 4]), что энтропия произвольного стационарного источника определена и удовлетворяет неравенствам $0 \le H(S) \le \log k$, где k — мощность алфавита. Для энтропии источника без памяти S справедлива формула Шеннона

$$H(S) = -\sum_{i=1}^{k} p(a_i) \log p(a_i).$$

Обозначим через $A^* = \cup_{n=1}^{\infty} A^n$ множество конечных слов в алфавите A. *Кодированием* называется инъективное отображение $f : A^* \to \{0, 1\}^*$. Двоичное слово $f(x)$ называется кодом слова $x \in A^*$. Через $|f(x)|$ будем обозначать длину кодового слова. *Префиксным* будем называть кодирование, обладающее следующим свойством: для произвольных слов $x, y \in A^*$ слово $f(x)$ является началом (префиксом) слова $f(y)$ только

тогда, когда слово x является началом слова y. Декодирование (вычисление функции f^{-1}) префиксного кода значительно упрощается, поскольку по началу кодового слова $f(x)$ можно определить начало закодированного слова x.

Стоимостью $C(f,S)$ кодирования f источника S называется среднее число битов, которое нужно затратить для кодирования одной буквы исходного сообщения, т.е.

$$C(f,S)=\limsup_{n\to\infty}\frac{1}{n}\sum_{x\in A^n}p(x)|f(x)|. \qquad (2)$$

Избыточностью кодирования f слова $x\in A^n$ называется величина

$$r(f,x)=|f(x)|-\log\frac{1}{p(x)}. \qquad (3)$$

Средней избыточностью $R(f,S)$ кодирования (на букву исходного сообщения) назвается разность между стоимостью кодирования (см. (2)) и энтропией источника (см. (1)), т.е.

$$R(f,S)=C(f,S)-H(S)=\limsup_{n\to\infty}\frac{1}{n}\sum_{x\in A^n}p(x)r(f,x). \qquad (4)$$

Известная теорема Шеннона (см., например, [1, 2, 3, 4]) утверждает, что

1. Для произвольного кодирования f и произвольного стационарного источника S средняя избыточность кодирования неотрицательна, т.е. $R(f,S)\geq 0$.

2. Для произвольного стационарного источника S и числа $\varepsilon>0$ найдется префиксное кодирование f_0 такое, что $R(f_0,S)<\varepsilon$.

Таким образом, энтропия источника является точной нижней гранью стоимости кодирования или числа двоичных символов, которое нужно в среднем затратить на кодирование одной буквы сообщения.

2. Описание метода кодирования

Одним из наиболее эффективных методов построения для данного стационарного источника префиксного кодирования со сколь угодно малой избыточностью является арифметическое кодирование. Основная идея арифметического кодирования состоит в следующем наблюдении. Все наборы из n букв алфавита A упорядочим лексикографически, т.е. $a_{i_1}a_{i_2}\dots a_{i_n}\prec a_{j_1}a_{j_2}\dots a_{j_n}$, если $i_k=j_k$ при $k<l$ и $i_l\leq j_l$.

Для каждого слова $x \in A^n$ определим величины $L(x) = \sum\limits_{y \prec x, y \in A^n} p(y)$ и $R(x) = L(x) + p(x)$.

Разделим полуинтервал $[0, 1)$ на полуинтервалы $[L(x), R(x))$. В каждом полуинтервале длины I, $I \leq 1$ найдется двоично-рациональное число со знаменателем не большим, чем $2^{\lceil -\log I \rceil}$, поскольку разность между любыми двумя ближайшими числами с таким знаменателем не превосходит I ($2^{-\lceil -\log I \rceil} \leq 2^{\log I} = I$). Каждому слову $x \in A^n$ поставим в соответствие двоично-рациональное число $q(x) \in [L(x), R(x))$ со знаменателем, не большим $2^{\lceil -\log p(x) \rceil}$. В качестве кода $f(x)$ рассмотрим двоичную запись числителя числа $q(x)$, которая имеет длину, равную $\lceil -\log p(x) \rceil$ (если число двоичных символов в записи числителя меньше, то нужно дописать слева недостающие нули), т.е.

$$|f(x)| = \lceil -\log p(x) \rceil. \tag{5}$$

Поскольку полуинтервалы $[L(x), R(x))$, соответствующие различным словам x одинаковой длины, не пересекаются, то все числа $q(x)$, $x \in A^n$ попарно различны, т.е. отображение $f : A^n \to E^*$ инъективно. Если для всех $x \in A^n$ и $a_i \in A$ справедиво неравенство $p(a_i|x) \leq 1/2$, то при добавлении каждой следующей буквы интервал уменьшается не менее чем в два раза. Тогда $|f(xa_i)| > |f(x)|$ и отображение $f : A^* \to E^*$ оказывается инъективным. Из формул (3)–(5) следует, что избыточность кодирования f произвольного слова $x \in A^*$ не превосходит единицы и средняя избыточность кодирования источника равняется нулю.

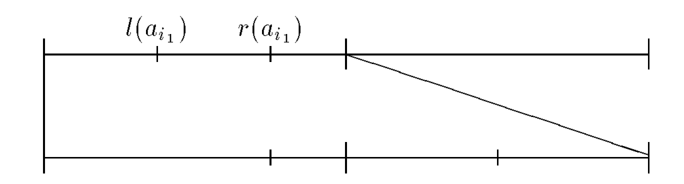
Однако в таком виде арифметическое кодирование не может применяться на практике, поскольку требуемая точность арифметических вычислений экспоненциально растет с увеличением длины кодируемого слова. Действительно, при увеличении длины слова на одну букву, число различных слов данной длины возрастает в k раз, а значит соответствующий слову полуинтервал уменьшается в среднем в k раз. Тогда для кодирования слова на единицу большей длины нужно в k раз точнее определять границы соответствующего полуинтервала. Ниже будет описан наиболее известный вариант арифметического кодирования, который использует арифметические вычисления с фиксированной точностью. Для упрощения описания метода положим, что S — источник Бернулли и $p(a_i) \leq 1/4$ для всех $1 \leq i \leq k$. Длина $t > 0$ двоичной записи чисел, с которыми в процессе кодирования выполняются арифметические операции, является параметром кодирования и должна удовлетворять неравенству $p(a_i) \geq 1/2^{t-2}$ для всех $a_i \in A$.

Определим величины $\sigma_1 = 0$, $\sigma_i = \sigma_{i-1} + p(a_{i-1})$. Разделим полуинтервал $[0, 1)$ на полуинтервалы $i(a_i) = [l(a_i), r(a_i))$, где $l(a_i) = \lfloor \sigma_i 2^t \rfloor / 2^t$

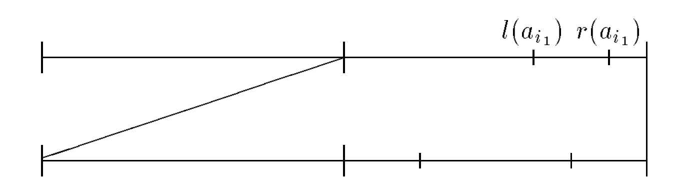
и $r(a_i) = \lfloor \sigma_{i+1} 2^t \rfloor / 2^t$. Каждой букве $a_i \in A$ соответствует i-й полуинтервал, длина которого приблизительно равна $p(a_i)$. Пусть $x = a_{i_1} a_{i_2} \ldots a_{i_n}$. Поскольку $p(a_i) \leq 1/4$, возможны три случая:
1) $i(a_{i_1}) \subset [0, 1/2)$;
2) $i(a_{i_1}) \subset [1/2, 1)$;
3) $i(a_{i_1}) \subset [1/4, 3/4)$.

Пусть $I(x)$ — полуинтервал, соответствующий слову $x \in A^n$. Тогда в первом случае имеем $I(x) \subset i(a_{i_1}) \subset [0, 1/2)$ и первым символом после запятой всех чисел $q \in I(x)$ является 0 . Поэтому 0 — первый символ кода $f(x)$. Во втором случае первым символом кода $f(x)$ является 1. В третьем случае первый символ пока неизвестен.

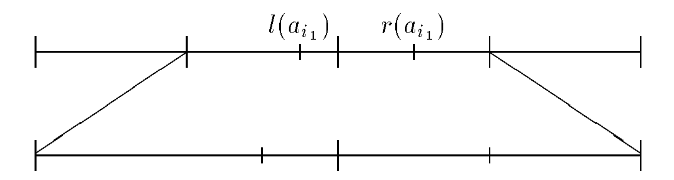
Проведем операцию изменения масштаба (растяжения) полуинтервала. В первом случае новый полуинтервал — $[2l(a_{i_1}), 2r(a_{i_1}))$.



Во втором случае — $[2l(a_{i_1}) - 1, 2r(a_{i_1}) - 1)$.



В третьем случае — $[2l(a_{i_1}) - 1/2, 2r(a_{i_1}) - 1/2)$.



В каждом случае новый полуинтервал длиннее полуинтервала $i(a_{i_1})$ в два раза. Будем продолжать процедуру изменения масштаба до тех пор, пока растянутый полуинтервал укладывается целиком в правую, левую или среднюю половину полуинтервала $[0,1)$. При растяжении левой половины полуинтервала $[0,1)$ будем приписывать 0 в качестве следующего символа кода $f(x)$, при растяжении правой половины $[0,1)$ будем приписывать 1. Если перед растяжением левой (правой) половины $[0,1)$ текущий полуинтервал оказывался h раз в средней половине, то после нуля (единицы) нужно приписать h единиц (нулей), т.е. в целом нужно приписать к коду слово $01\ldots1\,(10\ldots0)$.

В итоге получим полуинтервал $[l/2^t, r/2^t)$, где l и r — целые, $0 \le l < r \le 2^t$ и $r-l > 2^{t-2}$. Из $[l/2^t, r/2^t)$ выделим полуинтервал, соответствующий второй букве слова x: $i(a_{i_1}a_{i_2}) = [l(a_{i_1}a_{i_2}), r(a_{i_1}a_{i_2}))$, где $l(a_{i_1}a_{i_2}) = (l + \lfloor(r-l)\sigma_{i_2}\rfloor)/2^t$ и $r(a_{i_1}a_{i_2}) = (l + \lfloor(r-l)\sigma_{1+i_2}\rfloor)/2^t$. Проделаем с полуинтервалом $i(a_{i_1}a_{i_2})$ те же операции изменения масштаба, что и с полуинтервалом $i(a_{i_1})$. В результате получим начало кода $f(x)$, соответствующее двум начальным буквам. Выполнив такую же процедуру для букв $a_{i_3}, a_{i_4}, \ldots a_{i_n}$, получим код $f(x)$. Заметим, что при вычисления кода $f(x)$ мы использовали арифметические операции с числами, по модулю не превосходящими 2^t, причем для кодирования каждой очередной буквы достаточно конечного числа арифметических операций.

Из процедуры арифметического кодирования непосредственно вытекает следующее утверждение.

Утверждение 1. *Полуинтервалы, соответствующие словам одинаковой длины, не пересекаются. Соответствующий слову x полуинтервал целиком содержится в полуинтервале, соответствующем каждому началу слова x.*

В том случае, когда условие $\max p(a_i) \le 1/4$ не выполняется, операция изменения масштаба не обязательно следует за добавлением очередной буквы. Очередные буквы необходимо добавлять, пока соответствующий полуинтервал не будет целиком укладываться в правую, левую или среднюю половину полуинтервала $[0,1)$. Если вероятности букв зависят от предыстории, то при кодировании буквы a_{i_j} слова $x = a_{i_1}a_{i_2}\ldots a_{i_n}$ вместо чисел σ_i следует использовать числа σ_i^j, $1 \le j \le n$, определенные равенствами $\sigma_1^j = 0$, $\sigma_i^j = \sigma_{i-1}^j + p(a_{i-1}|a_{i_1}a_{i_2}\ldots a_{i_{j-1}})$.

Рассмотрим пример арифметического кодирования ($t=4$) слова $a_4a_1a_2$, порожденного источником Бернулли в алфавите $A = \{a_1, a_2, a_3, a_4, a_5\}$ и вероятностями появления букв $p(a_1) = p(a_3) = 1/4$, $p(a_2) = p(a_4) = p(a_5) = 1/6$.

Получаем равенства:
$l(a_4) = \frac{\lfloor 16\cdot 2/3\rfloor}{16} = \frac{10}{16}$, $r(a_4) = \frac{\lfloor 16\cdot 5/6\rfloor}{16} = \frac{13}{16}$,
$l(a_4a_1) = \frac{\lfloor 12\cdot 0\rfloor}{16} = 0$, $r(a_4a_1) = \frac{\lfloor 12/4\rfloor}{16} = \frac{3}{16}$,
$l(a_4a_1a_5) = \frac{\lfloor 12\cdot 5/6\rfloor}{16} = \frac{10}{16}$, $r(a_4a_1a_5) = \frac{\lfloor 12\cdot 1\rfloor}{16} = \frac{12}{16}$.
Процесс построения арифметического кода изображен на рис. 1. Получаем $f(a_4a_1a_5) = 1010101$.

Декодирование арифметического кода осуществляется следующим образом. Сначала нужно определить первую букву закодированного слова. Затем нужно удалить начальные символы кода, соответствующие первой букве, и определить вторую букву закодированного слова и т. д. Например, определим первую букву слова, имеющего код 1010101. Так как первым символом кодового слова является 1, то первой буквой закодированного слова является a_3, a_4 или a_5. Вторым символом кодового слова является 0, поэтому первой буквой закодированного слова является a_3 или a_4. Так как следующим символом кодового слова является 1, мы заключаем, что первой буквой закодированного слова является буква a_4. Процесс декодирования изображен на рис. 2.

После повторения процедуры кодирования буквы a_4 (см. пример выше) разделим полученный полуинтервал на части, соответствующие буквам a_1, a_2, a_3, a_4 , a_5, и определим вторую букву закодированного слова аналогичным образом.

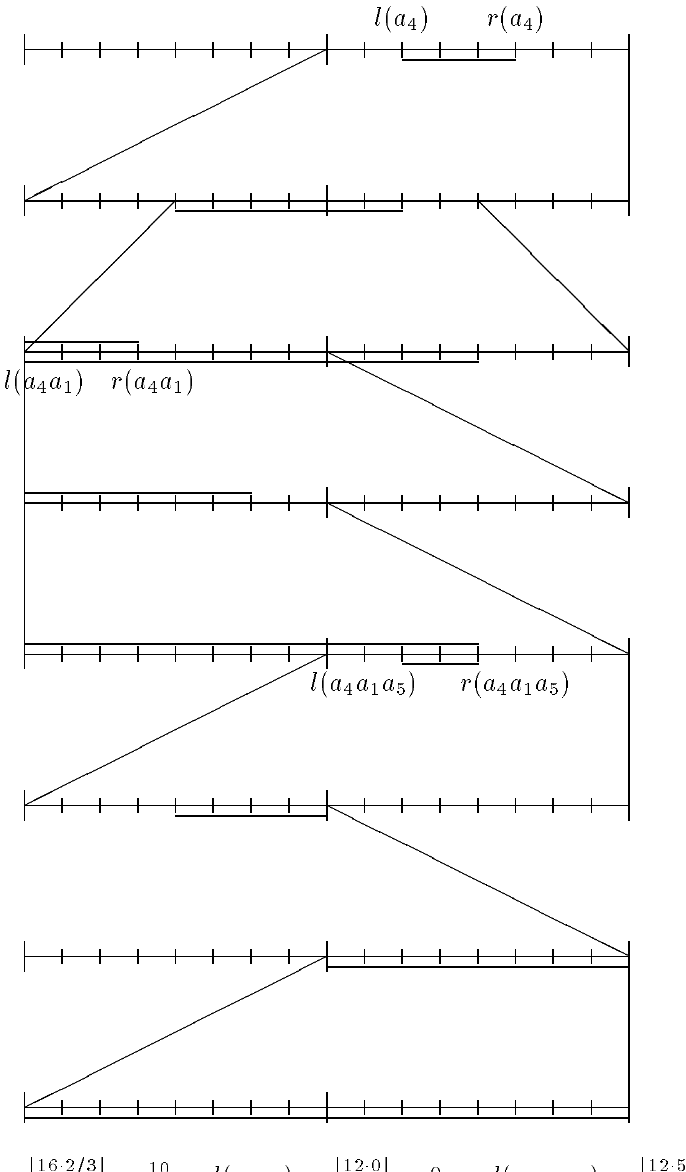
3. Избыточность арифметического кодирования

Запишем формально рекуррентные формулы, определяющие процедуру арифметического кодирования источника Бернулли. Пусть x — некоторое слово в алфавите A. Обозначим через xa_i слово, полученное из x добавлением буквы a_i. Пусть $d(x)$ — количество изменений масштаба при кодировании слова x; $L(x)$ и $R(x)$ — левая и правая границы полуинтервала, соответствующего слову x в первоначальном масштабе. Тогда арифметическое кодирование определяется следующими формулами: $R(\emptyset) = 1$, $L(\emptyset) = 0$, $d(\emptyset) = 0$,

$$I(x) = R(x) - L(x), \tag{6}$$
$$I(xa_i)2^{d(x)+t} = (\lfloor \sigma_{i+1} I(x) 2^{d(x)+t}\rfloor - \lfloor \sigma_i I(x) 2^{d(x)+t}\rfloor), \tag{7}$$
$$1/4 < I(x)2^{d(x)} \le 1, \tag{8}$$
$$L(xa_i)2^{d(x)+t} = L(x)2^{d(x)+t} + \lfloor \sigma_i I(x) 2^{d(x)+t}\rfloor. \tag{9}$$

В качестве кода слова x можно взять двоичную запись $\mathrm{Bin}_{d(x)+2}(N)$, состоящую из $d(x)+2$ символов, произвольного целого числа

$$N \in [L(x)2^{d(x)+2}, R(x)2^{d(x)+2}).$$



$l(a_4) = \frac{\lfloor 16 \cdot 2/3 \rfloor}{16} = \frac{10}{16}$ $\quad l(a_4a_1) = \frac{\lfloor 12 \cdot 0 \rfloor}{16} = 0$ $\quad l(a_4a_1a_5) = \frac{\lfloor 12 \cdot 5/6 \rfloor}{16} = \frac{10}{16}$

$r(a_4) = \frac{\lfloor 16 \cdot 5/6 \rfloor}{16} = \frac{13}{16}$ $\quad r(a_4a_1) = \frac{\lfloor 12/4 \rfloor}{16} = \frac{3}{16}$ $\quad r(a_4a_1a_5) = \frac{\lfloor 12 \cdot 1 \rfloor}{16} = \frac{12}{16}$

Рис. 1

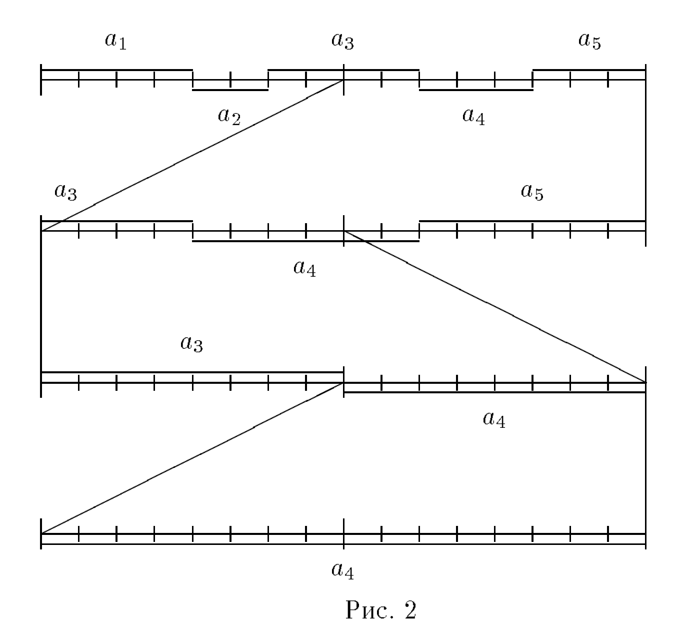


Рис. 2

Для определенности выберем число $L(x)2^{d(x)+2}$, т. е.

$$f(x) = \mathrm{Bin}_{d(x)+2}(L(x)2^{d(x)+2}). \quad (10)$$

Утверждение 2. *Пусть $A = \{a_1, \ldots, a_k\}$ — алфавит и при каждом i, $1 \le i \le k$, выполнены неравенства $1/2^{t-2} \le p(a_i) \le 1/4$. Тогда отображение f, определяемое формулами (6)–(10), является префиксным кодированием.*

Доказательство. Пусть $a, b \in E^*$ и a — префикс слова b. Тогда двоичные числа u и v, определенные равенствами $\mathrm{Bin}_{|a|}u = a$ и $\mathrm{Bin}_{|b|}v = b$, удовлетворяют неравенствам

$$u2^{-|a|} \le v2^{-|b|} < (u+1)2^{-|a|}. \quad (11)$$

Пусть x — произвольное слово в алфавите A. Из неравенства (8) следует, что

$$L(x) + 2^{-d(x)-2} \le L(x) + I(x) = R(x). \quad (12)$$

Докажем от противного, что отображение f обладает свойством префиксности. Пусть $x, y \in A^*$ — некоторые слова и слово $f(x)$ — префикс слова $f(y)$. Так как $|f(x)| = d(x) + 2$, то из (10)–(12) получаем

$$L(x) \le L(y) \le L(x) + 2^{-d(x)-2} \le R(x),$$

т.е. полуинтервалы $[L(x), R(x))$ и $[L(y), R(y))$ пересекаются. Из утверждения 1 вытекает, что $[L(y), R(y)) \subset [L(x), R(x))$ и слово x является префиксом слова y (противоположное включение противоречит тому, что кодовое слово $f(x)$ — префикс кодового слова $f(y)$). Следовательно, отображение f обладает свойством префиксности.

Докажем инъективность отображения f. Если $f(x) = f(y)$, то x — префикс слова y (или наоборот). Так как $\max p(a_i) \leq 1/4$, то справедливы неравенства $I(y) \leq I(xa_i) \leq I(x)/4$. Тогда из неравенства (8) следует, что $d(x) < d(y)$ и $|f(x)| < |f(y)|$. Таким образом, отображение f инъективно.

Ограничение $\max p(a_i) \leq 1/4$ в условии утверждения 2 можно снять, несколько усложнив кодирование конца сообщения.

Теорема 1. *Пусть S — такой источник Бернулли в алфавите $A = \{a_1, \ldots, a_k\}$, что при любом i, $1 \leq i \leq k$, выполнены неравенства $1/2^{t-3} \leq p(a_i) \leq 1/4$. Тогда для арифметического кодирования f с параметром t справедливо неравенство*

$$R(f, S) \leq \frac{\max 1/p(a_i)}{2^{t-3} \ln 2}.$$

Доказательство. Пусть $x = a_{i_1} \ldots a_{i_n}$. Введем обозначение $x^m = a_{i_1} \ldots a_{i_m}$ при $m \leq n$. Докажем методом индукции, что

$$I(x^n) > \prod_{j=1}^{n} (p(a_{i_j}) - 1/2^{t-2}). \tag{13}$$

Из равенства (7) следует, что

$$I(a_{i_1}) = \frac{1}{2^t}(\lfloor \sigma_{i_1+1} 2^t \rfloor - \lfloor \sigma_{i_1} 2^t \rfloor) > p(a_{i_1}) - 1/2^t.$$

Пусть неравенство (13) верно для $m-1$ при $m \leq n$. Тогда из (7) и (8) получаем

$$I(x^m) = I(x^{m-1} a_{i_m}) = \frac{1}{2^{d(x^{m-1})+t}}(\lfloor I(x^{m-1}) \sigma_{i_m+1} 2^{d(x^{m-1})+t} \rfloor -$$

$$- \lfloor I(x^{m-1}) \sigma_{i_m} 2^{d(x^{m-1})+t} \rfloor) > I(x_{m-1}) \left(p(a_{i_m}) - \frac{1}{I(x^{m-1}) 2^{d(x^{m-1})+t}} \right) \geq$$

$$\geq I(x_{m-1})(p(a_{i_m}) - \frac{1}{2^{t-2}}),$$

т.е. неравенство (13) доказано. Из (8) и (13) следует, что

$$1 > 2^{d(x)} \prod_{j=1}^{n} (p(a_{i_j}) - 1/2^{t-2})$$

откуда

$$d(x) < -\sum_{j=1}^{n} \log(p(a_{i_j}) - 1/2^{t-2}). \qquad (14)$$

Используя неравенство $\ln(1+x) \le x$ при $x > -1$, получаем

$$-\log(p(a_{i_j}) - 1/2^{t-2}) \le -\log p(a_{i_j}) + \frac{1}{\ln 2(p(a_{i_j})2^{t-2} - 1)}.$$

Тогда из (14) и условия теоремы $1/2^{t-3} \le p(a_i)$ следует неравенство

$$d(x) < -\sum_{j=1}^{n} \log p(a_{i_j}) + \frac{n \max 1/p(a_{i_j})}{2^{t-3} \ln 2} = -\log p(x) + \frac{n \max 1/p(a_{i_j})}{2^{t-3} \ln 2}.$$

Из формул (1)–(4), (10) и предыдущего неравенства получаем оценку средней избыточности кодирования f источника S

$$\begin{aligned} R(f,S) = C(f,S) - H(S) = \\ \limsup_{n\to\infty} \frac{1}{n} \sum_{x\in A^n} p(x)(d(x) + 2 - \log 1/p(x)) \le \\ \le \frac{\max 1/p(a_i)}{2^{t-3} \ln 2}. \end{aligned}$$

Теорема доказана.

Из теоремы непосредственно вытекает, что выбирая достаточно большой параметр t, можно получить кодирование со сколь угодно малой средней избыточностью кодирования R. Причем параметр t, равный длине двоичной записи чисел, с которыми в процессе кодирования нужно выполнять арифметические операции, растет только как $\log 1/R + const$ при $R \to 0$.

Аналогичная теорема об избыточности арифметического кодирования справедлива для широкого класса стационарных источников, как показано, например, в [5].

Литература

[1] Галлагер Р. Теория информации и надежная связь. М.: Советское радио, 1974.

[2] Кричевский Р. Е. Сжатие и поиск информации. М.: Радио и связь, 1989.

[3] Потапов В. Н. Теория информации. Кодирование дискретных вероятностных источников. Новосибирск: Изд. центр НГУ, 1999.

[4] Потапов В. Н. Обзор методов неискажающего кодирования дискретных источников // Дискрет. анализ и исслед. операций. Сер. 1. 1999. Т. 6, № 4. С. 49–91.

[5] Рябко Б. Я., Фионов А. Н. Эффективный метод арифметического кодирования для источников с большими алфавитами // Проблемы передачи информации. 1999. Т. 35, вып. 4. С. 95–108.

[6] Rissanen J. Generalized Kraft inequality and arithmetic coding// IBM J. Res. Develop. 1976. V. 20, № 3. P. 198–203.

[7] Rissanen J., Langdon G. Universal modelling and coding// IEEE Trans. Inform. Theory. 1981. V. IT-27, № 1. P. 12–23.

[8] Witten I. H., Neal R. M., Cleary J. G. Arithmetic coding for data compression// Commun. ACM. 1987. V. 30, № 6. P. 520–540.